

Abusive Text Classifier

A PROJECT REPORT

Submitted by

Ritika Sharma (24MCI10076)
Ravneet Singh (24MCI10070)
Lokesh Singh(24MCI10018)

In partial fulfillment for the award of the degree of

MASTERS OF COMPUTER APPLICATIONS

IN ARTIFICIAL INTELLIGENCE AND MACHINE

LEARNING



Chandigarh University

April 2025



BONAFIDECERTIFICATE

Certified that this project report “**Abusive Text Classifier**” is the Bonafide work of **Ritika Sharma (24MCI10076)** , **Ravneet Singh (24MCI10070)**, **Lokesh Singh(24MCI10018)** who carried out the project work under my/our supervision.

Dr. KRISHAN TULI

HEAD OF THE DEPARTMENT
(HOD)

UNIVERSITY INSITUTE OF
COMPUTING

JAVED ALAM

SUPERVISOR

Assistant Professor

UNIVERSITY INSTITUTE OF
COMPUTING

INTERNAL EXAMINER

EXTERNAL EXAMINER

Abusive Text Classifier

Objective

The primary objective of this project is to develop a machine learning-based abusive text classifier that can detect offensive or abusive language in user-provided text. The system aims to:

1. Identify abusive content using machine learning models such as Logistic Regression, Support Vector Machines (SVM), and Random Forest.
2. Highlight specific abusive words from the input sentence using a predefined list.
3. Provide an intuitive graphical user interface (GUI) for users to interact with the classifier and view predictions.

This tool is intended for use in applications such as social media moderation, comment filtering, and educational platforms to promote respectful communication.

Technology Used

The project leverages the following technologies:

Programming Language

- **Python:** The entire application is implemented in Python due to its extensive libraries for machine learning, natural language processing (NLP), and GUI development.

Libraries and Frameworks

1. Machine Learning

- **scikit-learn:** Used for training and evaluating Logistic Regression, SVM, and Random Forest classifiers.
- **Vectorizer:** Converts text data into numerical features using Term Frequency-Inverse Document Frequency (TF-IDF).

2. Data Processing

- **pandas:** For loading and preprocessing the dataset.
- **re:** For text cleaning and tokenization.

3. GUI Development

- `tkinter`: Provides an interactive graphical user interface for training models, entering text, and displaying results.

4. Metrics

- `accuracy_score`: To evaluate the performance of the models on test data.

Dataset

The system uses a dataset containing labeled examples of abusive and non-abusive text (`twitter_racism_parsed_dataset`). The dataset must include two columns: one for the text and another for the label

(0 for non-abusive, 1 for abusive).

Implementation

1. Dataset Preparation

- The dataset is loaded using `pandas` and preprocessed to ensure all text is in string format.
- A predefined list of abusive words (e.g., "idiot", "stupid", "moron") is used for keyword-based detection.

2. Text Vectorization

- Text data is converted into numerical features using `TfidfVectorizer`, which captures word importance in the context of the dataset.
- The vectorizer uses n-grams (unigrams, bigrams, trigrams) to capture contextual information.

3. Model Training

- Three machine learning models are trained:
 1. Logistic Regression
 2. Support Vector Machine (SVM)

3. Random Forest Classifier

- The models are trained on 80% of the dataset, with the remaining 20% used for testing.

4. Abusive Word Detection

- Input sentences are tokenized, and each word is checked against the predefined list of abusive words.
- If any abusive word is found, it overrides model predictions to classify the sentence as abusive.

5. Graphical User Interface (GUI)

- A GUI built with `tkinter` allows users to:
 - Train models by clicking a button.
 - Enter a sentence to classify as abusive or non-abusive.
 - View model predictions and detected abusive words.
 - Display model accuracies in a popup window.

6. Output Display

- Predictions from all three models are displayed in the GUI.
- If any model or keyword detection flags the sentence as abusive, a warning message is shown along with detected abusive words.

Result

The effectiveness of this system lies in combining:

- Supervised Machine Learning Models that learn patterns from labeled data using TF-IDF features.
- Keyword-based Detection which immediately flags texts containing offensive words.


Abusive Text Classifier

Train Models


you are fool

Predict

Logistic Regression: Abusive
SVM: Abusive
Random Forest: Abusive

 **ABUSIVE CONTENT DETECTED**

Detected abusive words: fool

 Consider replacing with more respectful language.

Show Model Accuracies

Abusive Text Classifier

Train Models

you are kind

Predict

Logistic Regression: Non-abusive
SVM: Non-abusive
Random Forest: Non-abusive

☒ All models agree: Non-abusive

Show Model Accuracies

Conclusion

The Abusive Text Classifier successfully integrates machine learning techniques with keyword-based detection to identify offensive language in user-provided text. Key takeaways include:

1. Strengths:

- Combines statistical machine learning with rule-based keyword detection for improved accuracy.
- Provides clear feedback through an intuitive GUI.
- Can be extended with additional abusive words or retrained on new datasets.

2. Limitations:

- The predefined list of abusive words may not cover all possible offensive terms.
- Models may misclassify sentences due to limitations in training data or lack of contextual understanding.

3. Future Improvements:

- Incorporate deep learning models like BERT for better contextual understanding.
- Expand the dataset with more diverse examples of abusive language.
- Add multilingual support for detecting abuse in multiple languages.